Measuring Similarity to Detect Qualified Links

Xiaoguang Qi, Lan Nie and Brian D. Davison
Department of Computer Science & Engineering
Lehigh University
{xiq204,lan2,davison}@cse.lehigh.edu

ABSTRACT

The early success of link-based ranking algorithms was predicated on the assumption that links imply merit of the target pages. However, today many links exist for purposes other than to confer authority. Such links bring noise into link analysis and harm the quality of retrieval. In order to provide high quality search results, it is important to detect them and reduce their influence. In this paper, a method is proposed to detect such links by considering multiple similarity measures over the source pages and target pages. With the help of a classifier, these noisy links are detected and dropped. After that, link analysis algorithms are performed on the reduced link graph. The usefulness of a number of features are also tested. Experiments across 53 query-specific datasets show our approach almost doubles the performance of Kleinberg's HITS and boosts Bharat and Henzinger's *imp* algorithm by close to 9% in terms of precision. It also outperforms a previous approach focusing on link farm detection.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

General Terms

Algorithms, Performance

Keywords

Web search engine, link analysis, link classification, web spam

1. INTRODUCTION

In modern web search engines, link-based ranking algorithms play an important role. Typical link analysis algorithms are based on the assumption that links confer authority. However, this assumption is often broken on the real web. As a result, the retrieval performance based on such naive link analysis is often disappointing. According to our experiments on more than fifty query-specific datasets, on average only four out of the top ten results generated by the HITS algorithm [15] are considered relevant to the query (details in Section 5).

The prevalence of links that do not (or should not) confer authority is an important reason that makes link analysis less effective. Examples of such links are links that are created for the purpose of advertising or navigation like those in Figure 1. These types of links are common on the Web. From a person's view, these links do carry some information that the authors of the web pages want to promote. However, from the perspective of link analysis algorithms, these links are noisy information because they do not show the authors' recommendation of the target pages. Traditional link analysis algorithms do not distinguish such noise from useful information. As a consequence, the target pages of these links could get unmerited higher ranking. Therefore, in order to provide better retrieval quality, the influence of such links needs to be reduced.

For years, researchers have been working on improving the quality of link analysis ranking, most often through improvements to the traditional PageRank [21] and HITS [15] algorithms. Some other work focuses on detecting and demoting web pages that do not deserve the ranking generated by traditional link analysis. However, little work has asked which links *should* be used in web link analysis.

In this paper, we introduce the notion of "qualified links"— links that are qualified to make a recommendation regarding the target page. We propose to detect qualified links using a classifier which, based on a number of similarity measures of the source page and target page of a link, makes the decision that whether the link is "qualified". After this, the "unqualified links" are filtered out, which leaves only the "qualified links". Link analysis algorithms are then performed on the reduced web graph and generate the resulting authority ranking. We also studied a number of features

Figure 1: Examples of links that do not confer authority

AIRWeb '07, May 8, 2007 Banff, Alberta, Canada.

in the "qualified link classification", revealing some interesting insights.

The contributions of this paper are:

- the novel notion of "qualified links" and a method to differentiate such links from those "unqualified";
- a study of the features being used to detect "unqualified links";
- an experimental comparison of our approach with other web ranking algorithms on real-world datasets.

The rest of this paper is organized as follows. The background of link analysis and related work in link spam detection and demotion is briefly reviewed in Section 2. Our motivation is presented in Section 3 and the methodology is detailed in Section 4. In Section 5, experimental results are presented. Finally, we conclude this paper with a discussion.

2. BACKGROUND AND RELATED WORK

The idea of incorporating link analysis in ranking algorithms was first considered a decade ago. In this section, we briefly review the background of link-based ranking algorithms and the related work in link spam detection.

2.1 Background

Kleinberg [15] proposed that web documents had two important properties, called hubness and authority, as well as a mechanism to calculate them. In his Hyperlink-Induced Topic Search (HITS) approach to broad topic information discovery, the score of a hub (authority) depended on the sum of the scores of the connected authorities (hubs):

$$A(p) = \sum_{q:q\to p} H(q) \quad \text{and} \quad H(p) = \sum_{q:p\to q} A(q)$$

Kleinberg calculated these scores on the subset of the web that included top-ranked pages for a given query, plus those pages that pointed to or were referenced by that set.

Bharat and Henzinger [2] proposed a number of improvements to HITS. One of the changes is an algorithm called *imp*, which re-weights links involved in mutually reinforcing relationships and drops links within the same host. They found that *imp* made a significant improvement over the original HITS.

Page and Brin [21, 3] proposed an alternative model of page importance, called the random surfer model. In that model, a surfer on a given page i, with probability $(1-d)$ chooses to select uniformly one of its outlinks $O(i)$, and with probability d to jump to a random page from the entire web W. The PageRank score for node i is defined as the stationary probability of finding the random surfer at node i. One formulation of PageRank is

$$PR(i) = (1-d)\sum_{j:j\to i}\frac{PR(j)}{O(j)} + d\frac{1}{N}$$

PageRank is a topic-independent measure of the importance of a web page, and must be combined with one or more measures of query relevance for ranking the results of a search.

2.2 Related work

In this paper we are concerned with the classification of hyperlinks regarding their usefulness in web link analysis. Thus, prior work in this area is quite relevant. Davison [8] first proposed the automatic recognition of nepotistic links—links that are present for reason other than merit. While that work considered seventy-five features, only a few dealt with content, looking at page titles and meta descriptions, and overlapping outgoing link sets. Rather than binary features, our current work focuses on a few similarity measures, and additionally considers the full content of the pages. Benczur et al. [1] proposed to detect nepotistic links using language models. In this method, a link is down-weighted if its source and target page are not related based on their language models. This approach is based on the assumption that pages that are connected by non-nepotistic links must be sufficiently similar, which is not required in our model. Chakrabarti et al. [5] extend HITS by increasing the weights of links whose anchor text (or surrounding text) incorporates terms from the query. Our approach does not examine text specifically in or around the anchor, and more importantly, is not query-specific.

While we focus on content, more work has considered the analysis of link structure to eliminate or down-weight links (e.g., to combat web spam). Here we consider a representative set of such work. Lempel and Moran [16] defined a tightly-knit community (TKC) as a small but highly connected set of sites. Even though such a community is not quite relevant to the query, it may still be ranked highly by link-based ranking algorithms. The authors proposed SALSA, a stochastic approach for link structure analysis, and demonstrated that it is less vulnerable to the TKC effect than HITS. Li et al. [17] pointed out the small-in, large-out link problem with HITS, in which a community is associated with a root with few in-links but many out-links. Such communities may dominate HITS results even if they are not very relevant. The authors addressed this problem by assigning appropriate weights to the in-links of root. Wu and Davison [25] proposed a two-step algorithm to identify link farms. The first step generates a seed set based on the intersection of in-link and out-links of web pages. The second step expands the seed set to include pages pointing to many pages within the seed set. The links between these identified spam pages are then re-weighted and a ranking algorithm is applied to the modified link graph. Carvalho et al. [7] also proposed algorithms to detect noisy links at site level by examining the link structure among web sites.

Finally, we note that many other approaches to web spam detection have been explored. This includes Drost and Scheffer's work on spam identification [9], the work by Fetterly et al. [10] and Ntoulas et al. [19] on spam detection using statistical analysis (of links and content), and Gyöngyi et al.'s work on using trust to demote spam [11]. These approaches typically focus on the identification of specific pages that should be labeled as spam rather than the links between them.

3. MOTIVATION

The early success of link-based ranking algorithms was predicated on the assumption that links imply merit of the target pages. However, in many instances this assumption is no longer valid. An evident example is spam links—links that are created for the sole purpose of manipulating the ranking algorithm of a search engine. The presence of link spam makes link analysis less effective. Another example is navigational links, where links are created for easy access to other pages regardless of relevance. Links between different regional web sites of the same company (`http://foobar.com/` and `http://foobar.co.uk/`), and links to the "terms of use" of a web site can be considered as examples of navigational links. Although navigational links are not created for the purpose of spamming, they should also be considered less valuable for link analysis since they hardly imply authority of their target pages.

Based on this motivation, we introduce the notion of "a qualified link". A qualified link is a link on a page that is qualified to make

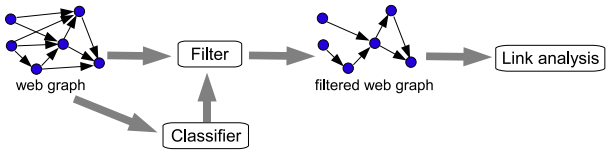

Figure 2: The process of "qualified" link analysis

a recommendation regarding the target page, and is in some sense the opposite of a nepotistic link [8].

Besides spam links and navigational links, other types of "unqualified links" include advertising links and irrelevant links. Advertising links are links created for the purpose of advertising. Irrelevant links can be considered as the collection of other "unqualified links", such as links pointing to a required document viewer or to a particular web browser for the desired display of the web page.

To determine whether a link is qualified or not, we propose to build a binary classifier based on the characteristics of links. Then based on the decision of that classifier, a filtering process can be performed on the web graph which retains only the "qualified" links. Finally, link analysis algorithms can run on the reduced graph and generate rankings.

Alternatively, if the classifier is able to generate a reasonable probability of a link being "qualified", link analysis may be performed on a weighted graph where each edge in the web graph is weighted by its relative qualification.

There are a variety of metrics one might use to measure the qualification of a link. The measures we use in this work are the similarity scores of the source page and the target page, such as content similarity and URL similarity. The similarity measures are detailed in Section 4. The classifier considers these similarity measures, and makes a decision (of a link being "qualified" or not) or predict a probability (of how likely a link being "qualified"). The process of this approach is visualized in Figure 2.

4. QUALIFIED LINK ANALYSIS

4.1 Similarity measures

A variety of features could be used to measure the qualification of a link. However, considering the issue of computational complexity, it is desirable to use a small number of features and to use features that are easy to compute. We propose predicting a link being "qualified" or not by considering the similarity scores of its source and target pages. Six features are used in this work; they are host similarity, URL similarity, topic vector similarity, tfidf content similarity, tfidf anchor text similarity, and tfidf non-anchor text similarity. The computation of these similarity measures are detailed as follows.

- **Host similarity.** Inspired in part by Kan and Thi [14], the host similarity of two web pages is measured by the portion of common substrings of the host names of the two web page URLs. Suppose s is a string and r is an integer, $Sub(s, r)$ is the set of all substrings of s with length r, $host_x$ is the host name of a web page x, then the host similarity of two web pages x and y is calculated by the Dice coefficient [24] of the two host names as shown in Equation 1.

$$Sim_{host}(x, y) = \frac{2 * |Sub(host_x, r) \cap Sub(host_y, r)|}{|Sub(host_x, r)| + |Sub(host_y, r)|} \quad (1)$$

  In the experiments of this work, r is set to 3.

- **URL similarity.** Analogous to host similarity, the URL similarity is measured by the common substrings that the URLs of two web pages have. Still using the notations above and suppose URL_x is the URL of web page x, then the URL similarity of two web pages x and y is calculated by Equation 2.

$$Sim_{URL}(x, y) = \frac{2 * |Sub(URL_x, r) \cap Sub(URL_y, r)|}{|Sub(URL_x, r)| + |Sub(URL_y, r)|} \quad (2)$$

  Here, r is also set to 3.

- **Topic vector similarity.** The topic vector similarity reflects how similar the topics of the two web pages are. If there are n pre-defined topics t_1 through t_n, then each web page x can be represented by a probability distribution vector $v_x = (v_{x,1}, v_{x,2}, ..., v_{x,n})$, in which each component $v_{x,i}$ is the probability that page x is on topic t_i. Such a vector can be obtained by various means. In our experiments, the topic vectors are computed using a naive Bayes classifier based on Rainbow [18]. Each component of a topic vector corresponds to a top-level category of ODP directory [20]. The topic vector similarity is computed as the product of the topic vectors of the two pages.

$$Sim_{topic}(x, y) = \sum_{i=1}^{n} v_{x,i} \times v_{y,i} \quad (3)$$

- **Tfidf content similarity.** The tfidf content similarity of two web pages measures the term-based similarity of their textual content. We use the equations used by the Cornell SMART system [23] to compute the tfidf representation of a web document. Given a collection D, a document $d \in D$, a term t, suppose $n(d, t)$ is the number of times term t occurs in document d, D_t is the set of documents containing term t, then the term frequency of term t in document d is

$$TF(d, t) = \begin{cases} 0 & \text{if } n(d, t) = 0 \\ 1 + log(1 + log(n(d, t))) & \text{otherwise} \end{cases} \quad (4)$$

  The inverse document frequency is

$$IDF(t) = log\frac{1 + |D|}{|D_t|} \quad (5)$$

  In vector space model, each document d is represented by a vector in which each component d_t is its projection on axis t, given by

$$d_t = TF(d, t) \times IDF(t) \quad (6)$$

  Then the content similarity of web pages x and y is computed as the cosine similarity of their vector space representations.

$$Sim_{content}(x, y) = \frac{\sum_{t \in T}(x_t * y_t)}{\sqrt{\sum_{t \in T} x_t^2} \cdot \sqrt{\sum_{t \in T} y_t^2}} \quad (7)$$

- **Anchor text similarity.** The anchor text similarity of two pages measures the similarity of the anchor text in those two pages. It is computed the same way as content similarity, except substituting each document by a "virtual document" consisting of all the anchor text inside that document. Still, the similarity score is computed as the cosine similarity of the two vectors, each representing a "virtual document". IDF is estimated on the collection of these "virtual documents".

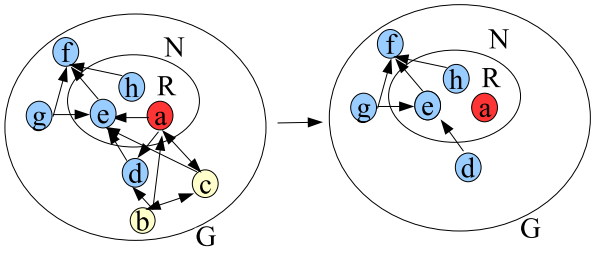

Figure 3: Pruning a query-specific graph.

- **Non-anchor text similarity.** The non-anchor text similarity of two pages measures the similarity of textual content that is not anchor text in those two pages. It is computed the same way as content similarity, except substituting each document by a virtual document consisting of all the textual content inside that document that is not anchor text. IDF is estimated on the collection of the "virtual documents".

4.2 Qualified HITS

Before introducing Qualified HITS, we first analyze the traditional HITS algorithm and discuss its drawbacks. HITS uses a two-step process to collect a query-specific dataset. The goal is to produce a small collection of pages likely to contain the most authoritative pages on a given topic. Starting from a given query, HITS assembles an initial collection of pages, typically, up to 200 top ranked pages returned by a text search engine on that query. Although this root set R is rich in relevant documents, it is typically restricted to those pages containing the query string. For most short queries, especially those representing a broad topic, such a limitation may exclude some strong authorities. In addition, there are often extremely few links between pages in R [15], rendering it essentially "structureless" and hard for later link analysis. To solve the problem, an expansion step is evoked from the root set. Consider a relevant page for the query topic, although it may well not be in the set R, it is quite likely to know or to be known by at least one page in R. Hence, the dataset is augmented by adding any pages that are linked to or from a page in the root set R. These interconnected candidates are then analyzed by the HITS algorithm to identify the best authorities.

However, both the dataset collection process and HITS analysis take the "links imply relevancy" for granted. Since they treat all the hyperlinks equally, they are vulnerable to "unqualified links". Irrelevant pages may dominate the query-specific web graph and ruin the ranking result. These unqualified hyperlinks break the relevance assumption, prevent the dataset from staying on the query topic, and bring noise to the HITS calculation as well. To solve this problem, we propose a simple heuristic approach to eliminate unqualified links and irrelevant pages from the dataset, which is introduced below as Qualified HITS.

Suppose we produce a focused web graph $G(V, E)$ for a given query using the HITS process described above, where V is the set of web pages and E represents hyperlinks among those pages. In addition, V consists of the initial root set R and the set of R's neighboring pages N. We then use the following rules to filter out noise in the graph G. An example is given in Figure 3.

- For every hyperlink in E, compute the similarity scores of its source page and target page. Feed these scores to a classifier which is trained on some labeled links. If the classifier gives a negative answer, we consider this hyperlink to be unqualified and should be removed from the graph. In the example, let us suppose links $a \to d$, $a \to e$, $c \to e$, $b \to d$, $b \to a$, $a \to c$ and $c \to a$ are removed.
- Scan the graph with unqualified links eliminated and check each page in the neighboring set N to see if it is still connected with the root set R. If the answer is negative, it is indicated that the page is not relevant to any page in the root set and should not be included in the data set in the beginning. As a result, this page, as well as all the links associated with it, are removed from the link graph. Back to the example, neighboring pages b and c are no longer connected with the root set R and thus are removed, as well as the links between them. d, f and g remain since they are still connected to the root set.

In summary, originally in this example, pages a, b, c and d form a densely-connected community and dominate the link graph. After the two steps above, the graph is converted from the one on the left to the right one in Figure 3. As a result, the connectivity inside this community is reduced and some irrelevant pages are directly removed. The reputation of these pages are thus successfully demoted. On the other hand, those good authorities, such as f and e are not affected much.

4.3 Qualified PageRank

The method of qualified PageRank is the same as qualified HITS except that the second step is unnecessary since PageRank runs on the global link graph as opposed to a query-specific graph.

5. EXPERIMENTS

5.1 Datasets

Qualified-HITS needs to be tested on query-specific datasets. In order to evaluate Qualified-HITS, we used the query-specific datasets collected by Wu and Davison [25]. The corpora includes 412 query-specific datasets, with 2.1 million documents. The queries are selected from the queries used in previous research, the category name of ODP directory, and popular queries from Lycos and Google.

The HITS dataset collecting process was used for each query; Yahoo! was queried to get the top 200 URLs; then for each URL, the top 50 incoming links to this URL are retrieved by querying Yahoo! again. All pages referenced by these top 200 URLs were also downloaded. Query specific graphs were generated by parsing the retrieved web pages. Intra-host links were eliminated.

From this dataset, we used the same twenty queries as Wu and Davison and then randomly selected an additional 38 queries, and used the combined 58 query-specific datasets to evaluate the performance of Qualified-HITS. These queries are shown in Table 1.

In their work on link spam detection, they presented a two-step algorithm for detecting link farms automatically. As a result, spam pages are identified and the links among them are dropped (or down-weighted).

Qualified PageRank is evaluated on a 2005 crawl from the Stanford WebBase [6], which contained roughly 58 million pages and 900 million hyperlinks.

5.2 Human labeling of links

In order to build a classifier which categorizes links into qualified links and unqualified links, a set of labeled training data is needed. We manually labeled 1247 links that were randomly selected from five query-specific datasets (marked with ** in Table 1). To each link, one of the following labels was assigned: recommendation, navigational, spam, advertising, irrelevant, and undecidable. These

california lottery(**)	table tennis(**)	weather(**)
aerospace defence(**)	IBM research center(**)	
image processing(*)	rental car(*)	healthcare(*)
jennifer lopez(*)	super bowl(*)	web proxy(*)
art history(*)	teen health(*)	trim spa(*)
translation online(*)	web browser(*)	wine(*)
US open tennis(*)	hand games(*)	picnic(*)
online casino	IT company	music channel
source code download	humanities	wall street
native+tribal	theatre	morning call
kids entertainment	library	mtv download
education reference	party games	local search
ask an expert	gifts shopping	stocks
music shopping	pets shopping	E-commerce
business service	small business	rebate online
Chinese web portal	wholesale	food drink
healthcare industry	chemicals	tennis games
mental health	addictions	TV channel
health insurance	dentistry	car buying
breaking news	weblog news	

Table 1: Queries used for collecting query-specific data sets.

labels are not directly used to train the classifier. Instead, they are mapped to two labels, qualified and unqualified. Recommendation links are considered qualified, while, navigational, spam, advertising, and irrelevant links are unqualified. A link is labeled undecidable if the content of its source or target page is not available. This category of links is not used to train the classifier.

Two human editors (the first two authors) were involved in this labeling task. In order to estimate how consistent their decisions are, their individual labeling results on 100 links are compared. On 85 links, their decisions are the same. After mapping the labels to qualified or unqualified, they agree on 94 links. This comparison does not only reflect the consistency of the labeling, but also provides a rough upper bound on how well the classifier could do.

5.3 Link classification

Based on the human-labeled links, a linear SVM classifier is trained and tested using SVM^{light} [13]. The 1016 labeled samples (undecidable links are excluded) are randomly split into two halves, on which a two-fold cross validation is performed. The average accuracy is 83.8%. The precision and recall of positive class (qualified links) are 71.7% and 82.2%, respectively. The trained model shows that anchor text similarity is the most discriminative feature, followed by non-anchor text similarity.

To find out how discriminative the anchor text similarity is, we trained and tested a linear SVM classifier on the anchor text similarity only. The average accuracy is 72.8%, significantly lower than that using all the six features.

For comparison, to estimate the upper bound of classification performance, we trained a classifier on the whole labeled set and tested the training accuracy. The accuracy is 85.1%, with precision and recall being 73.7% and 83.4%.

In order to get better insight into the features, we plot the human-assigned labels to feature values in six graphs (Figure 4 through Figure 9), each showing one of the features. For each feature, the possible range of the feature values is equally divided into 20 sub-ranges (or, buckets). In each graph, x-axis depicts the set of value ranges. The bar graph shows the distribution of that feature of all human-labeled links. The line graph shows the percentage of qualified links in each range.

From Figure 4, we can see that the distribution of topic vector similarity is somewhat polarized, with the majority gathering at the first and last range. This is because the topic vector given by the

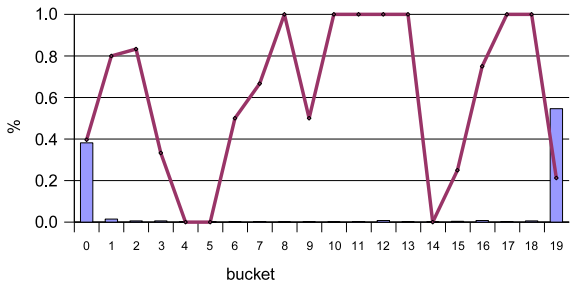

Figure 4: Topic vector similarity

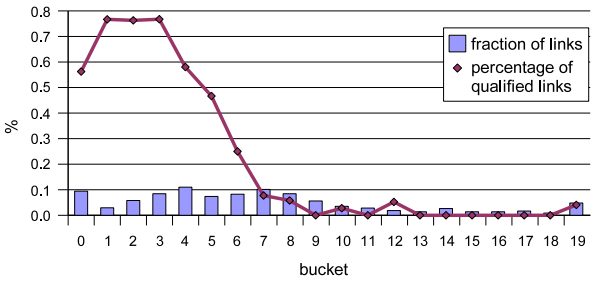

Figure 5: Content similarity

textual classifier is polarized. In most vectors, one component dominates others. As a result, the cosine similarity of two vectors tend to be quite close to zero or one. The fluctuation of the probability of "qualified links" indicates that topic vector similarity is not a good feature for detecting "qualified links".

Compared with the distribution of topic vector similarity, the distributions of the three content based features (content similarity, anchor text similarity, and non-anchor text similarity, shown in Figure 5, Figure 6, and Figure 7, respectively) are more smooth. About the probability of "qualified links", although there are still minor fluctuations, the high probability within the first three or four buckets, followed by a dramatic decrease from the fifth to seventh bucket, shows that the links in the rear buckets are mostly "unqualified links". This result indicates that links between two pages that are too similar are likely to be "unqualified". This matches our observation in practice on navigational links and spam links, where the source and target pages often have a large portion of content or anchor text in common.

The results on host name similarity and URL similarity, shown in Figure 8 and Figure 9, are not so interesting. They are easy to compute, but their usefulness here is also limited.

5.4 Retrieval performance of Qualified HITS

The classification of links is only an intermediate step. The final goal of qualified link analysis is to improve retrieval performance. Here, we test Q-HITS on the query-specific datasets, and compare its result with that of Bharat and Henzinger's *imp* algorithm [2]. Since five of the query-specific datasets have been used for human labeling of links, the remaining 53 query-specific datasets are used for the evaluation of retrieval performance. A linear SVM classifier, trained on all the human-labeled links, is used to classify the links within the query-specific datasets. 23% of the 1.1 million links are classified as "unqualified" by the classifier and removed. Then the *imp* algorithm is applied to the reduced graph to generate the results for each query.

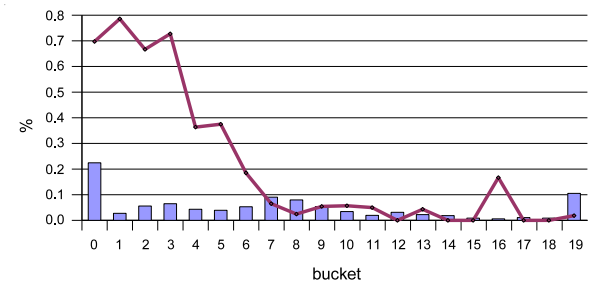

Figure 6: Anchor text similarity

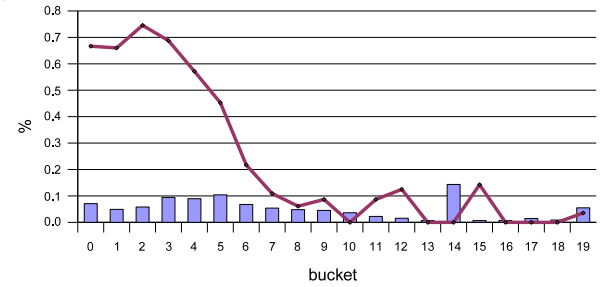

Figure 7: Non-Anchor text similarity

Since there is no available evaluation for results of these query-specific datasets, the relevance between query and search results have to be inspected manually. In our evaluation system, the top ten search results generated by various ranking algorithms are mixed together. To evaluate the performance, 43 participants were enlisted, to whom a randomly chosen query and a randomly selected set of ten results (of those generated for the given query) were shown. The evaluators were asked to rate each result as quite relevant, relevant, not sure, not relevant, or totally irrelevant, which were internally assigned the scores of 2, 1, 0, -1, -2, respectively. A page is marked as relevant if its average score is greater than 0.5.

Based on the evaluation data, we can calculate the overall precision at 10 (P@10) for each approach; in addition, the overall average relevance score (S@10) is calculated to further explore the quality of retrieval since precision cannot distinguish high-quality results from merely good ones. We also evaluated the ranking algorithms over the Normalized Discounted Cumulative Gain (NDCG) [12] metric. NDCG credits systems with high precision at top ranks by weighting relevant documents according to their rankings in the returned search results; this characteristic is crucial in web search.

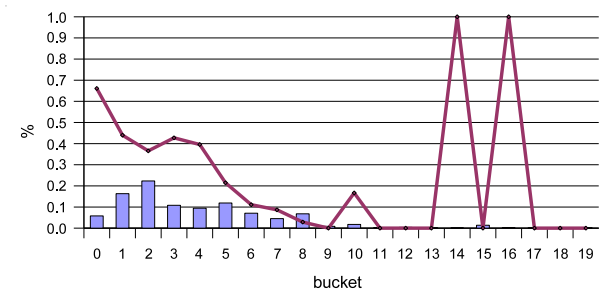

Figure 8: Host name similarity

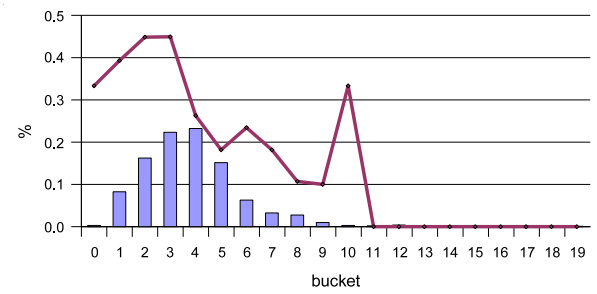

Figure 9: URL similarity

We used these metrics to compare the performance of the different approaches.

5.4.1 *Sample search results*

Here we first demonstrate the striking results this technique makes possible by an example query "US open tennis". In Table 2, the top 10 results returned by *imp* are dominated by a group of touring and traveling pages that are strongly connected. After applying Q-HITS, the links inside this community is broken. The undeserved authority of its members are reduced.

5.4.2 *Evaluation*

Figure 10 shows the comparison of original HITS, *imp*, and Q-HITS. The average precision of the top 10 results (precision@10) of HITS is only 0.38. *imp* improved that by a large difference to 0.69. By filtering out "unqualified links", precision@10 can be further improved to 0.75; the average score is improved by almost one third from 0.74 to 0.96, compared to *imp*. T-tests show that the

Rank	URL
1	http://www.luxurytour.com/
2	http://www.rivercruisetours.com/
3	http://www.escortedtouroperators.com/
4	http://www.atlastravelweb.com/
5	http://www.atlastravelnetwork.com/
6	http://www.sportstravelpackages.com/
7	http://www.atlasvacations.com/
8	http://www.escortedgrouptours.com/
9	http://www.escorteditalytours.com/
10	http://www.atlascruisevacations.com/

(a) Top 10 results by *imp*

Rank	URL
1	http://www.tennis.com/
2	http://www.usopen.org/
3	http://www.wtatour.com/
4	http://www.usta.com/
5	http://www.atptour.com/
6	http://www.itftennis.com/
7	http://www.frenchopen.org/
8	http://www.gotennis.com/
9	http://www.tennistours.com/
10	http://www.sportsline.com/u/tennis/

(b) Top 10 results by Q-HITS

Table 2: Results for query *US open tennis*.

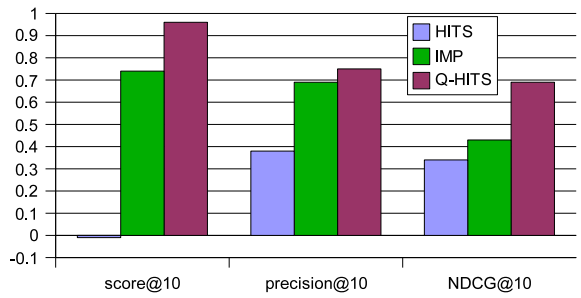

Figure 10: Retrieval performance on 53 query-specific datasets

improvement of Q-HITS over *imp* in precision and score are both statistically significant (with p-values of 0.024 and 0.012, respectively).

We also compared our approach with the link farm detection work by Wu and Davison [25] (denoted as "Link farm removal") on the 15 queries in common (marked with * in Table 1). The result is shown in Figure 11. On those 15 query-specific datasets, the precision@10 of HITS is 0.30. Link farm removal boosts that to 0.65. Having a precision@10 of 0.78, Q-HITS outperforms "Link farm removal" by 20%.

In "Link farm removal" algorithm, the links among identified link farm members are dropped. We compared the links dropped by Q-HITS (i.e., unqualified links according to the classifier) and the links that are dropped by "Link farm removal". Q-HITS dropped 37.17% of all the links; "Link farm removal" dropped 18.93%. The intersection of the links drop by the two algorithms accounts for 17.30% of all the links, showing that Q-HITS generates close to a superset of dropped links.

5.5 Retrieval performance of Qualified PageRank

We applied Qualified PageRank (Q-PR) on the WebBase dataset and compared its retrieval performance with PageRank (PR). The queries used in the experiment is listed in Table 3. Again, the SVM classifier trained on all the human-labeled links is used to classify the 900 million links. This time, only 0.4% of of the links are classified as "unqualified". After that, PageRank is performed on the reduced matrix generating the static ranking. The final result for each query is generated by an order-based linearly weighted combination of the static ranking and OKAPI BM2500 [22] weighting function (.8 for PageRank). The parameters of BM2500 equation are set the same as in [4].

Figure 12 shows the experimental result of Qualified PageRank and PageRank. We can see that dropping a tiny portion of link

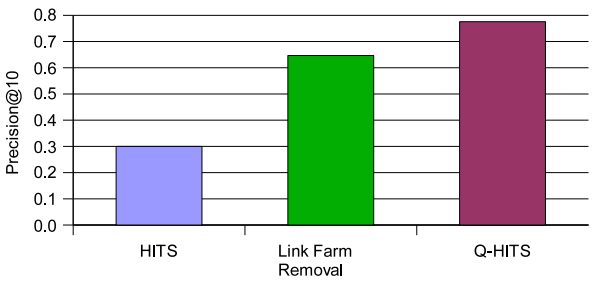

Figure 11: Retrieval performance on 15 query specific datasets

tsunami	diamond bracelet	windshield wiper
brad pitt	music lyrics	weight watchers
games	britney spears	halloween costumes
diabetes	olsen twins	automobile warranty
iraq war	college football	new york fireworks
madonna	harry potter	lord of the rings
poker	jennifer lopez	herpes treatments
playstation	jersey girl	the passion of christ
poems	george w. bush	musculoskeletal disorders
tattoos	online dictionar	st patricks day cards

Table 3: Queries used to test Qualified PageRank.

(0.4%) is able to increase the precision from 0.58 to 0.60 (with score@10 increased from 0.64 to 0.68).

6. DISCUSSION AND CONCLUSION

In this paper, we presented the approach of identifying qualified links by computing a number of similarity measures of their source and target pages. Through experiments on 53 query-specific datasets, we showed that our approach improved precision by 9% compared to the Bharat and Henzinger *imp* variation of HITS.

This paper is merely a preliminary study, demonstrating the potential of our approach. The following limitations can be addressed in future work.

- The classifier and similarity measures being used are quite simple. It is expected that the use of a better classification algorithm and an advanced set of similarity measures would produce a better result. For example, examining the similarity of text in and around a link to its target (as in [5]) might fare better, especially for multi-topic hubs.
- The punishment of removing "unqualified links" might be too stringent. A manual examination of the experimental results revealed that some authoritative pages are removed in addition to poor quality pages. Weighting the links by their quality could be a better alternative than the current binary weighting.
- The computational complexity of "qualified link analysis" is an issue that requires careful consideration. Although the index of the corpus could be made available before hand, computing the similarity scores is still expensive considering the size of the web. Potential solutions include using fewer features, using features that are easy to compute, and utilizing simple classification algorithms. We tested one possible extension, which builds a thresholding classifier based on anchor text similarity. The classifier simply categorizes the links within the first eight buckets as "qualified links", and the rest as "unqualified". This approach gives a precision of

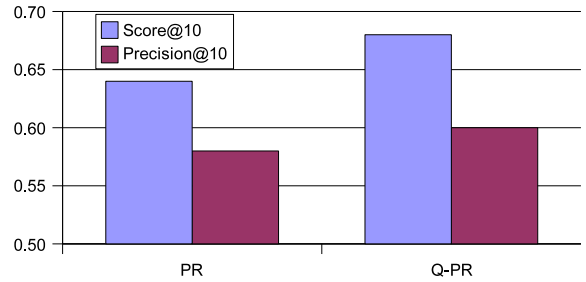

Figure 12: Retrieval performance of Q-PR on WebBase dataset

negative class ("unqualified links") at 97.05% on the labeled training set. This classifier is then applied to the 53 query-specific datasets. The retrieval performance is between that of *imp* and Q-HITS (precision@10 being 73.02%, score@10 being 0.93).

- As mentioned in Section 5.1, the data collection process used an existing search engine, which could introduce certain bias into the dataset. Building a dataset from a large web crawl can solve such a problem.
- The experiment testing Qualified PageRank is still rudimentary. In order to fully examine its usefulness, more experimental work on global datasets is needed.
- This approach is not a panacea for "unqualified links". We did not differentiate the different types of "unqualified links". Some types of links are perhaps more difficult to identify than others. Finer-grained discrimination might further boost retrieval quality. As a preliminary investigation in this direction, we trained a multi-class classifier to distinguish each individual type of "unqualified link". The result showed that the classifier was effective in finding spam links, while not very helpful in finding other types of "unqualified links". This remains a topic for future study.

Acknowledgments

We thank Baoning Wu for helpful discussions and providing the query-specific datasets and Stanford University for access to their WebBase collections. This work was supported in part by Microsoft Live Labs and the National Science Foundation under awards IIS-0328825 and IIS-0545875.

7. REFERENCES

[1] A. A. Benczur, I. Biro, K. Csalogany, and M. Uher. Detecting nepotistic links by language model disagreement. In *Proceedings of the 15th WWW conference*, pages 939–940, New York, NY, USA, 2006. ACM Press.

[2] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, Aug. 1998.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, pages 107–117, Brisbane, Australia, Apr. 1998.

[4] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, July 2004.

[5] S. Chakrabarti, B. E. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. M. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World Wide Web Conference*, pages 65–74, Brisbane, Australia, Apr. 1998.

[6] J. Cho, H. Garcia-Molina, T. Haveliwala, W. Lam, A. Paepcke, S. Raghavan, and G. Wesley. Stanford WebBase components and applications. *ACM Transactions on Internet Technology*, 6(2):153–186, 2006.

[7] A. L. da Costa Carvalho, P. A. Chirita, E. S. de Moura, P. Calado, and W. Nejdl. Site level noise removal for search engines. In *Proceedings of the 15th WWW conference*, pages 73–82, New York, NY, USA, 2006. ACM Press.

[8] B. D. Davison. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, pages 23–28. AAAI Press, July 2000. Presented at the AAAI-2000 workshop on Artificial Intelligence for Web Search, Technical Report WS-00-01.

[9] I. Drost and T. Scheffer. Thwarting the nigritude ultramarine: learning to identify link spam. In *Proceeding of the ECML*, 2005.

[10] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of WebDB*, pages 1–6, June 2004.

[11] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, 2004.

[12] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000.

[13] T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[14] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In *Proceeding of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 325–326, 2005. Poster abstract.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[16] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1–6):387–401, 2000.

[17] L. Li, Y. Shang, and W. Zhang. Improvement of HITS-based algorithms on web documents. In *Proceedings of the 11th International World Wide Web Conference*, pages 527–535. ACM Press, 2002.

[18] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow, 1996.

[19] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on the World Wide Web*, Edinburgh, Scotland, May 2006.

[20] Open Directory Project (ODP), 2007. `http://www.dmoz.com/`.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Unpublished draft, 1998.

[22] S. E. Robertson. Overview of the OKAPI projects. *Journal of Documentation*, 53:3–7, 1997.

[23] G. Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Retrieval*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[24] C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979. 2nd edition.

[25] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference*, pages 820–829, Chiba, Japan, May 2005.